

Class 7: Machine Learning 1

Zoe Matsunaga (PID: A16853288)

Table of contents

Clustering	1
K-means	4
Hierarchical CLustering	7
Principal Component Analysis (PCA)	9
Data Import	9
PCA to the rescue	12
Interpreting PCA Results	14

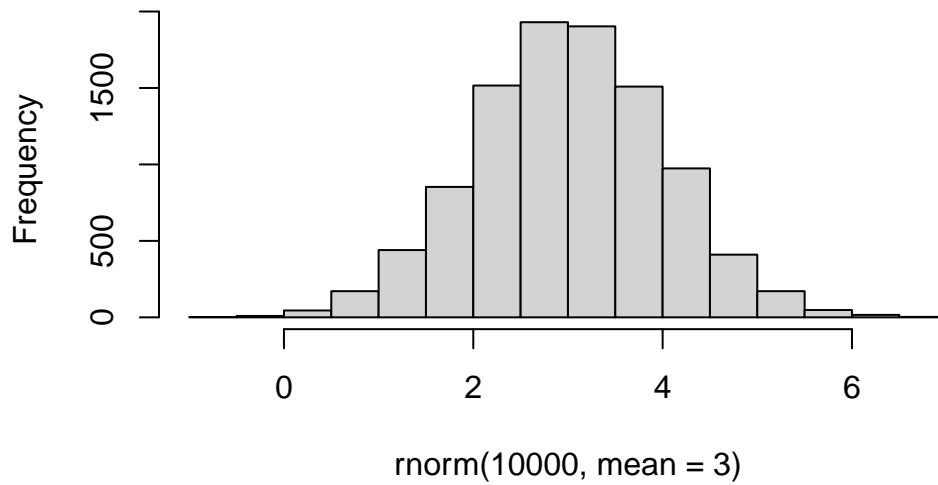
Today we will explore unsupervised machine learning methods starting with clustering and dimensionality reduction.

Clustering

To start, let's make up some data using the `rnorm()` function to cluster where we know what the answer should be.

```
hist(rnorm(10000, mean = 3))
```

Histogram of rnorm(10000, mean = 3)



Return 30 numbers centered on -3 and +3

```
tmp <- c(rnorm(30, mean=-3),  
        rnorm(30, mean =3))  
  
x <- cbind(x=tmp, y=rev(tmp))  
  
x
```

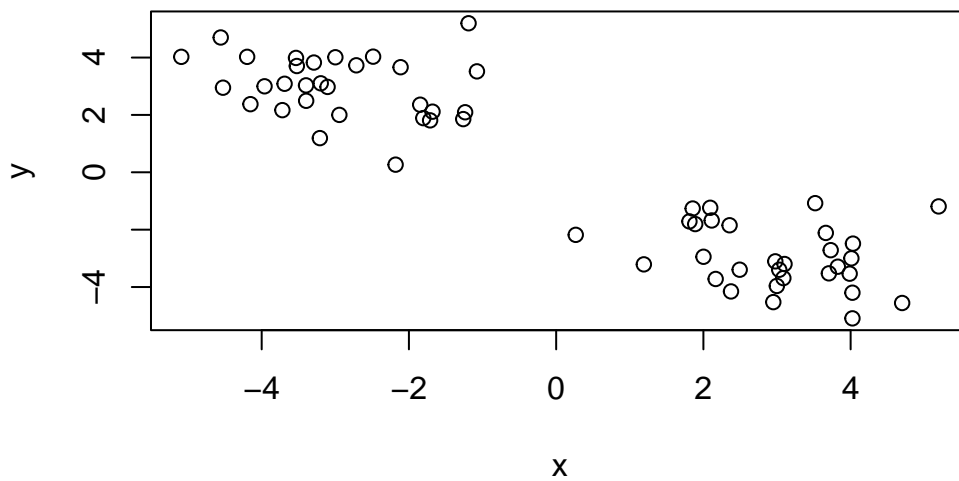
	x	y
[1,]	-1.2624580	1.8543780
[2,]	-3.1048777	2.9764749
[3,]	-1.2364936	2.0926067
[4,]	-3.3957916	2.4931010
[5,]	-3.9612493	2.9977720
[6,]	-1.1898618	5.1953104
[7,]	-1.8467059	2.3556872
[8,]	-2.1131427	3.6627124
[9,]	-1.0760893	3.5182008
[10,]	-4.5565243	4.6997019
[11,]	-5.0922045	4.0244377
[12,]	-3.7190927	2.1665539
[13,]	-2.9992774	4.0099133

[14,]	-4.5256077	2.9493111
[15,]	-4.1975633	4.0243539
[16,]	-2.9449636	2.0012660
[17,]	-3.6890747	3.0859966
[18,]	-3.3976309	3.0301118
[19,]	-3.5226596	3.7046149
[20,]	-3.5339208	3.9868209
[21,]	-2.1807950	0.2661589
[22,]	-3.2095534	1.1911264
[23,]	-1.7122489	1.8103656
[24,]	-2.4865773	4.0314871
[25,]	-1.8050173	1.8887158
[26,]	-2.7145046	3.7292546
[27,]	-1.6795467	2.1129569
[28,]	-3.1977667	3.1019300
[29,]	-4.1530950	2.3749713
[30,]	-3.2910301	3.8256596
[31,]	3.8256596	-3.2910301
[32,]	2.3749713	-4.1530950
[33,]	3.1019300	-3.1977667
[34,]	2.1129569	-1.6795467
[35,]	3.7292546	-2.7145046
[36,]	1.8887158	-1.8050173
[37,]	4.0314871	-2.4865773
[38,]	1.8103656	-1.7122489
[39,]	1.1911264	-3.2095534
[40,]	0.2661589	-2.1807950
[41,]	3.9868209	-3.5339208
[42,]	3.7046149	-3.5226596
[43,]	3.0301118	-3.3976309
[44,]	3.0859966	-3.6890747
[45,]	2.0012660	-2.9449636
[46,]	4.0243539	-4.1975633
[47,]	2.9493111	-4.5256077
[48,]	4.0099133	-2.9992774
[49,]	2.1665539	-3.7190927
[50,]	4.0244377	-5.0922045
[51,]	4.6997019	-4.5565243
[52,]	3.5182008	-1.0760893
[53,]	3.6627124	-2.1131427
[54,]	2.3556872	-1.8467059
[55,]	5.1953104	-1.1898618
[56,]	2.9977720	-3.9612493

```
[57,] 2.4931010 -3.3957916
[58,] 2.0926067 -1.2364936
[59,] 2.9764749 -3.1048777
[60,] 1.8543780 -1.2624580
```

Make a plot of x

```
plot(x)
```



K-means

The main function in “base” R for K-means clustering is `kmeans()`:

```
km <- kmeans(x, centers = 2)
km
```

K-means clustering with 2 clusters of sizes 30, 30

Cluster means:

	x	y
1	2.972065	-2.926511

2 -2.926511 2.972065

Clustering vector:

[illegible]

Within cluster sum of squares by cluster:

```
[1] 70.12433 70.12433
      (between_SS / total_SS =  88.2 %)
```

Available components:

```
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

The `kmeans()` function return a “list” with 9 components. You can see the named components of any loist with the `attributes()` function.

```
attributes(km)
```

\$names

```
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

```
$class
```

```
[1] "kmeans"
```

Q. How many points are in each cluster?

```
km$size
```

```
[1] 30 30
```

Q. Cluster assignment/membership vector?

```
km$cluster
```

[illegible]

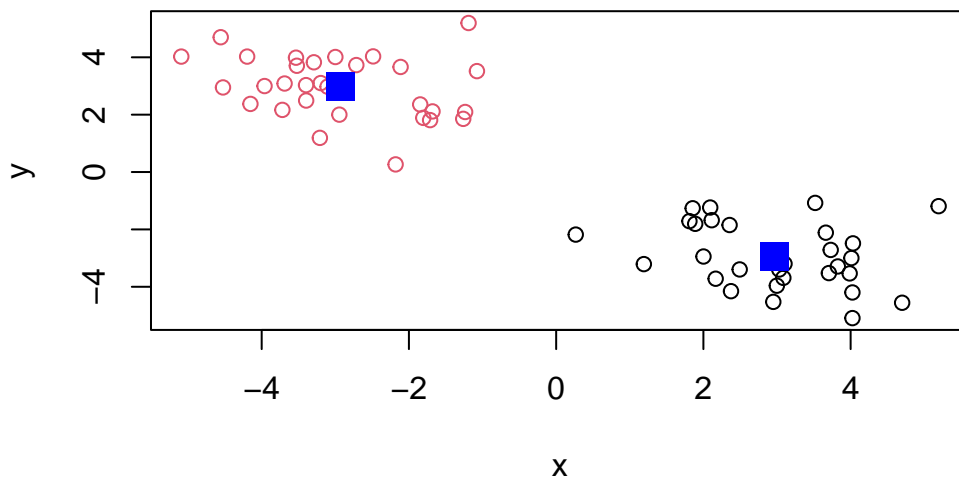
Q. Cluster centers?

```
km$centers
```

	x	y
1	2.972065	-2.926511
2	-2.926511	2.972065

Q. Make a plot of our `kmeans()` results showing cluster assingment using different colors for each group/cluster of points and cluster centers.

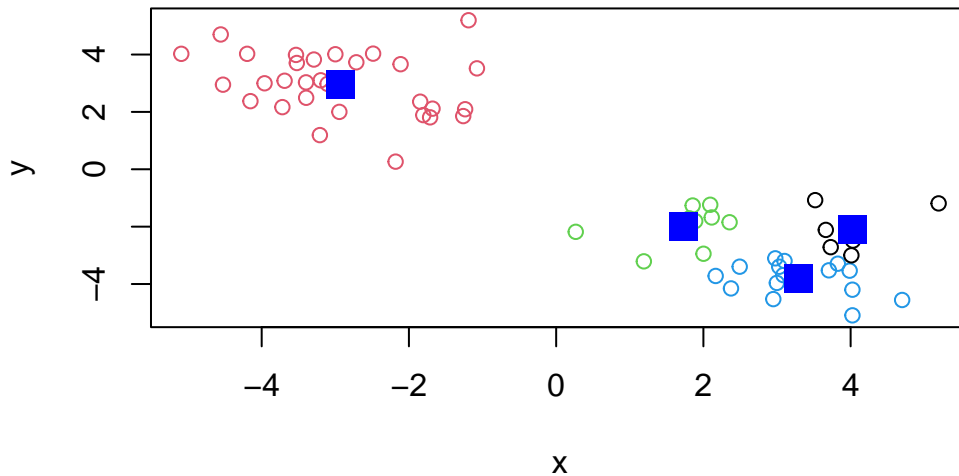
```
plot(x, col=km$cluster)
points(km$centers, col="blue", pch=15, cex=2)
```



Q. Run `kmeans()` again on `x` and this cluster into 4 groups/clusters and plot the same result figure as above.

```
km4 <- kmeans(x, centers=4)

plot(x, col=km4$cluster)
points(km4$centers, col="blue", pch=15, cex=2)
```



Key-point: K-means clustering is super popular, but can be misused. One big limitation is that it can impose a clustering pattern on your data even if clear, natural grouping doesn't exist - i.e. it does what you tell it to do in terms of centers.

Hierarchical CLustering

The main function in “base” R for hierarchical clustering is called `hclust()`.

You can't just pass our dataset as is into `hclust()`. You must give a “distance matrix” as input. We can get this from the `dist()` function in R.

```
d <- dist(x)
hc <- hclust(d)
hc
```

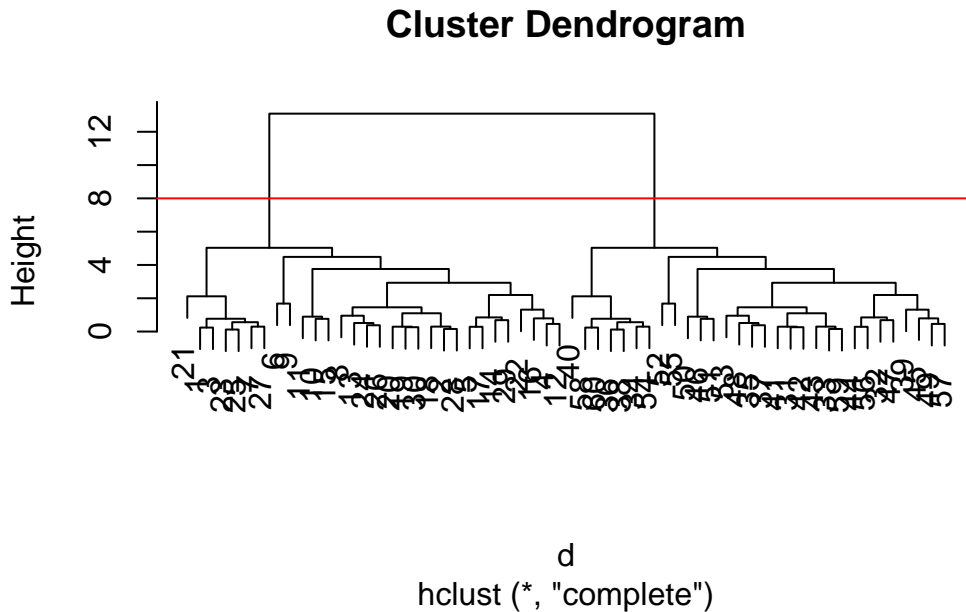
Call:

```
hclust(d = d)
```

```
Cluster method : complete
Distance       : euclidean
Number of objects: 60
```

The results of `fhclust()` don't have a useful `print()` method, but do have a special `plot()` method.

```
plot(hc)
abline(h=8, col="red")
```



To get our main cluster assignment (membership vector) we need to “cut” the tree at the big goal posts...

```
grps <- cutree(hc, h=8)
grps
```

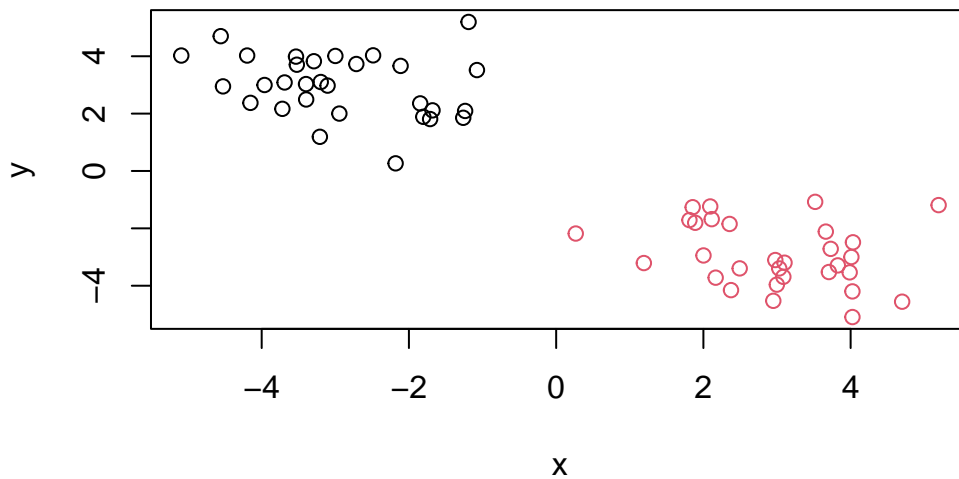
[1] 1 2 2 2 2 2 2 2 2
[39] 2

```
table(grps)
```

```
grps
  1  2
30 30
```



```
plot(x, col=grps)
```



Hierarchical clustering is distinct in that the dendrogram (tree figure) can reveal the potential group in your data, unlike K-means.

Principal Component Analysis (PCA)

PCA is a common and useful dimensionality reduction technique used in many fields - particularly bioinformatics.

Here we will analyze some data from the UK on food consumption.

Data Import

```
url <- "https://tinyurl.com/UK-foods"
x <- read.csv(url)

head (x)
```

	X	England	Wales	Scotland	N.Ireland
1	Cheese	105	103	103	66
2	Carcass_meat	245	227	242	267
3	Other_meat	685	803	750	586
4	Fish	147	160	122	93
5	Fats_and_oils	193	235	184	209
6	Sugars	156	175	147	139

```
rownames(x) <- x[,1]
x <- x[,-1]
head(x)
```

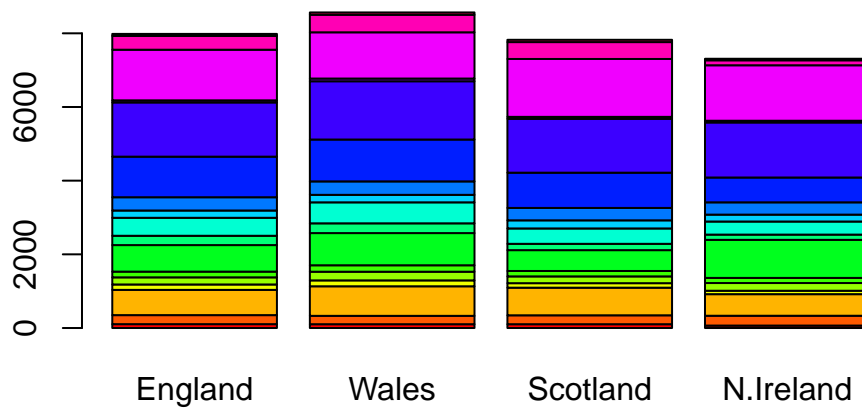
	England	Wales	Scotland	N.Ireland
Cheese	105	103	103	66
Carcass_meat	245	227	242	267
Other_meat	685	803	750	586
Fish	147	160	122	93
Fats_and_oils	193	235	184	209
Sugars	156	175	147	139

```
#destructive!
```

```
x <- read.csv(url, row.names=1)
head(x)
```

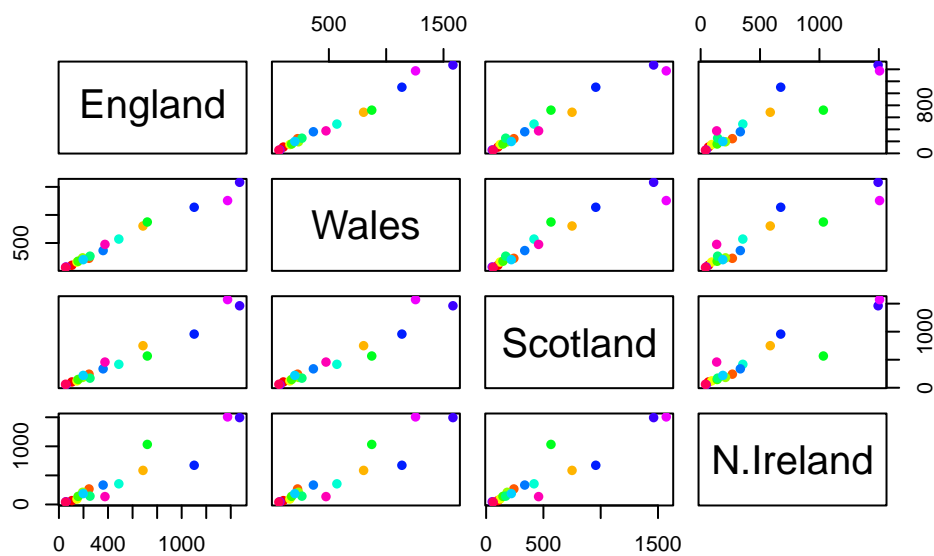
	England	Wales	Scotland	N.Ireland
Cheese	105	103	103	66
Carcass_meat	245	227	242	267
Other_meat	685	803	750	586
Fish	147	160	122	93
Fats_and_oils	193	235	184	209
Sugars	156	175	147	139

```
barplot(as.matrix(x), beside=F, col=rainbow(nrow(x)))
```



One conventional plot that can be useful is called a “pairs” plot.

```
pairs(x, col=rainbow(nrow(x)), pch=16)
```



PCA to the rescue

The main function in “base” R for PCA is `prcomp()`.

```
pca <- prcomp( t(x) )  
summary(pca)
```

Importance of components:

	PC1	PC2	PC3	PC4
Standard deviation	324.1502	212.7478	73.87622	2.921e-14
Proportion of Variance	0.6744	0.2905	0.03503	0.000e+00
Cumulative Proportion	0.6744	0.9650	1.00000	1.000e+00

The `prcomp()` function returns a list object of our results with five attributes/components.

```
attributes(pca)
```

```
$names  
[1] "sdev"      "rotation" "center"   "scale"    "x"
```

```
$class  
[1] "prcomp"
```

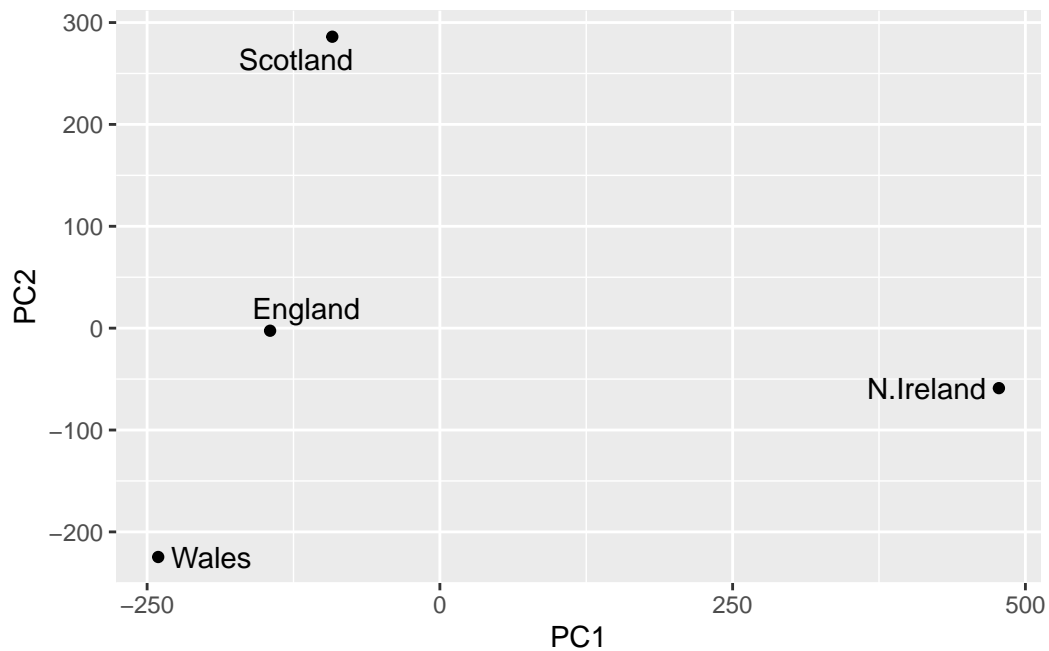
The two main “results” in here are `pca$x` and `pca$rotation`. The first of these (`pca$x`) contains the scores of the data on the new PC axis - we use these to make our “PCA plot”.

```
pca$x
```

	PC1	PC2	PC3	PC4
England	-144.99315	-2.532999	105.768945	-9.152022e-15
Wales	-240.52915	-224.646925	-56.475555	5.560040e-13
Scotland	-91.86934	286.081786	-44.415495	-6.638419e-13
N.Ireland	477.39164	-58.901862	-4.877895	1.329771e-13

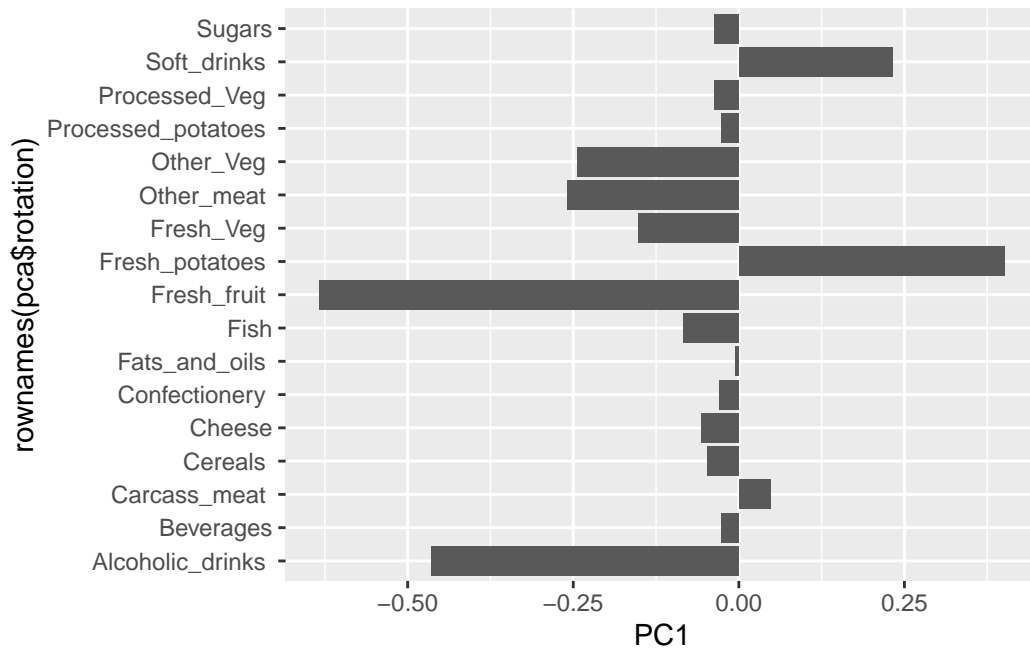
```
library(ggplot2)  
library(ggrepel)  
  
#Make a plot of pca$x with PC1 vs PC2
```

```
ggplot(pca$x) +
  aes(PC1, PC2, label=rownames(pca$x)) +
  geom_point() +
  geom_text_repel()
```



The second major result is contained in the `pca$rotation` object or component. Let's plot this to see what PCA is picking up...

```
ggplot(pca$rotation) +
  aes(PC1, rownames(pca$rotation)) +
  geom_col()
```



Interpreting PCA Results

This plot shows that for PC1 there are associations that can be made between certain food groups and the four European countries. The most amount of variance can be seen in categories, such as “Fresh_Fruit” leaning towards England, and “Fresh_Potatoes” favoring Northern Ireland.